pyrefra manual

Program reads a number of seg2 or segy input files and allows certain data treatment like different filters and picking with different methods. Different plotting options are given like gaining, shot/ receiver/ distance gathers. Files may be found at https://github.com/HZeyen/PyRefra.

Thanks to my son Manuel for creating the PyRefra icon.

Installation

1. **Mouse**
   It is strongly recommended to use a mouse. The program needs the central mouse button (or wheel). If you are working with a touchpad, configure it such that it simulates the central mouse button. The simplest is certainly to configure a three-finger-touch as central button. For this, under Windows press the WINDOWS-Key+I. On the screen that appears, click on “Périphériques” (the English version may be "Devices"). There, on the left-side panel, search “Pavé tactile” ("Touch Pad" in English?). Scroll down until you see the image with three fingers. Below this image, under “Appuis”, change to “Button central de la souris” ("Central mouse button"?).

2. Download and install **Anaconda Individual Edition**
3. Push the Windows key and search Anaconda3. There, open **Anaconda Prompt.** This opens a command window. Then type the following commands:

   ```
   conda config --add channels gimli --add channels conda-forge
   ```

   « channels » is the place where to find the source code, here « gimli » and “conda-forge”

   Pygimli is often not up to date concerning the last Python version. In this case, you are obliged to install a specific environment to be able to run Pyrefra with the tomography option. To simplify the installation manual, we suppose that this is the case. Create first the environment “pg” (pg is any name you want to give to the environment. If you want to call it, e.g., refra, change pg to refra in the following two commands):

   ```
   conda create -n pg pygimli
   ```

   Change focus to this new environment:

   ```
   conda activate pg
   ```

   Install pyrefra together with the necessary libraries:

   ```
   pip install pyrefra
   ```

   For manual install, you need the following libraries:

   ```
   conda install obspy statsmodels colorcet datetime scikit-learn spyder
   ```

   For **MAC**, it seems one must give the minimum pygimli version:

   ```
   conda –n pg –c gimli -c conda-forge "pygimli>=1.5.3"
   conda activate pg
   pip install pyrefra
   ```

If it is impossible to install pyGimli, you may continue without it, but the tomography option will not be available.

Then you may **modify a few default settings in some packages** for nicer plots or to avoid warnings. The corresponding edited files may be found on the github page (https://github.com/HZeyen/PyRefra/tree/main) with the same path structure as in the pyGIMLi and obspy site-packages in Anaconda.

To find the place where all packages are installed, write from within the Anaconda Powershell prompt:

```
python -c "import sysconfig; print(sysconfig.get_path('purelib'))"
```

This gives on my Windows installation: C:\Users\Hermann\anaconda3\envs\pg\Lib\site-packages

I will call this folder "ENV" in what follows.

You may simply copy the folders pygimli and obspy into the ENV folder. The other option, if you do not want to copy-paste the provided folders, is to modify the following files:

For nicer plots in **pyGIMLi**, modify numbering format in **drawDataMatrix**
The function is found in file **ENV/pygimli/viewer/mpl/dataview.py**
There, search lines starting with

ax.set_xticklabels and
ax.set_yticklabels

and change the rounding to 0 ciphers instead of the default 2 ciphers:

ax.set_xticklabels(['{:g}'.format(round(xx[int(ti)], **0**)) for ti in xt])

In addition, it is interesting, though not absolutely necessary to add the following code to pgimly which allows summing up the number of rays and ray lengths having crossed each cell during all iterations and in this way plotting the final model avoiding the areas where no ray at all passed during any of the iteration steps:

File **ENV/pygimli/frameworks/inversion.py**
After line 1378 of version1.5.3 (containing `startModel = self.startModel`**) add (outside the if):**

```
self.cov_sum = np.zeros(len(self.fop.startModel()))
self.cell_rays = []
```

After line 1466 (starting with `self.modelHistory.append`**) add:**

```
self.cov_sum += self.fop.jacobian().transMult(
    np.ones(self.fop.jacobian().rows()))
self.cell_rays.append([])
for i in range(len(self.cov_sum)):
    n = len(np.where(np.array(self.fop.jacobian().col(i)) != 0)[0])
    self.cell_rays[-1].append(n)
```

In **obspy**, you may find file seg2.py (usually in **ENV/obspy/io/seg2/seg2.py**)

There find line "**`if 'DELAY' in header['seg2']:`**" and comment out the whole "if" block. (a comment in python is the “#” character at the beginning of a line)
Also at the very end, find the line containing "**`WARNING_HEADER`**" and comment it out.
This is not important for working of the program, but it avoids an annoyingly long warning message each time you start the program.

On the other hand, in the same file, we detected a problem which appears on acquisition systems using anglophone date coding, i.e. mm/dd/yyyy instead of the expected dd/mm/yyyy. A work-around, only to get the program reading these data and not crashing has been programmed in lines 183-188 with a try-except construction which gives the correct date if the day of month is larger than 12, but confounds day and month for smaller days:

Instead of the two lines

```
day, month, year = int(date[0]), MONTHS[date[1].lower()], \
    int(date[2])
```

Write:

```
try:
    day, month, year = int(date[0]), MONTHS[date[1].lower()], \
        int(date[2])
except KeyError:
    day, month, year = int(date[1]), MONTHS[date[0].lower()], \
        int(date[2])
```

If under Linux, you get this error message: AttributeError: 'numpy.int64' object has no attribute 'split', go to file **ENV/obspy/util/misc.py**, near line 217 and replace “`except TypeError:`” by “`except (TypeError, AttributeError):`” (N.B.: since version 1.4.1, this problem seems to be solved)

Preparation

The program is based on the files produced by DMT stations like Summit2 and Summit_X. These two systems have in common that (at least with our configuration), no coordinates are stored in the file / trace headers. Therefore, the program requires two geometry files:

shots.geo

File contains four columns

Shot point number (as recorded during acquisition)
X_shot [m]
Y_shot [m]
Z_shot [m] (positive upwards)

In this file, every shot point number should have different coordinates associated. If due to recording errors two shots at the same coordinates were recorded with different shot point numbers, use file “file_corrections.dat” to correct shot point number for the corresponding file(s) (see below).

receivers.geo

File contains the same kind of information as shots.geo, also in 4 columns. However, a fifth column may be added containing any letter describing the geophone component recorded in a trace (e.g., when working with 3-component geophones). You may select traces by this letter. Usually, letters are V or Z for vertical component, E or N for geographic directions, L or T for longitudinal or transverse components. However, any other letters are allowed. By default (no 5th column in a line), the letter "Z" is associated to a trace. So, you are not obliged to write the letter into all lines.

If your system does not write the header words RECEIVER_STATION_NUMBER and SOURCE_STATION_NUMBER which are the ones corresponding to the first columns of files receivers.geo and shots.geo, the following definitions are made: RECEIVER_STATION_NUMBER is set to header keyword CHANNEL_NUMBER. SHOT_STATION_NUMBER is set to the number of your file (see below: NNNNN). If this is not the good numbering, use file "file_corrections.dat" to change numbers. See below for explanations.

The program plots traces generally as function of offset which is calculated using the three coordinates. Therefore, in principle, absolute metric coordinates (e.g., UTM) may be given in both geometry files. However, we suggest strongly to convert coordinates to a local system and **eliminate a possible trend**, such that the line goes in X or Y direction. The program, internally, uses only the coordinate which has the largest extent (important for cdp points used, e.g, for distance gathers or export to SEGY format).

Data file names: Files acquired with Summit equipment have by default the structure prefixNNNNN.ext. "prefix" is set by the user during acquisition. NNNNN is file number always with 5 ciphers filled to the left with zeros. "ext" may be "sg2" (Summit 2) or "seg2" (Summit X1). Also segy files are read with extension "sgy" or "segy".

If the file names have a different structure, make sure that the file number is always just before the dot. The number of ciphers may be variable, not necessarily filled to the left with zeros.

Optionally, there may be two files that allow **correction of certain acquisition errors**:

file_corrections.dat

This file is used to correct general errors for recorded files. You may only indicate files for which there are really problems. The file has six columns:

irec, ipos, ir_start, ir_step, dt, interp

irec: Number of record (file number)
DMT files have names of the following form:
prefixnnnnn.sg2 for Summit 2 or prefixnnnnn.seg2 for Summit_X
prefix may be user-defined, often "rec" or "shot_"
nnnnn is shot number (not shot-point number! The same shot point may be recorded in different files, if shots are repeated) in 5 ciphers, filled with zeros to the left.
Example: recorded files have the names [01000.sg2, 01100.sg2,01200.sg2]. Then irec should be [1000, 1100, 1200] (not [0, 1, 2])

ipos: Number of shot point position (counting starts with SP 1, not 0, i.e., natural numbering)
If ipos is <= 0, actual shot point number is left unchanged

ir_start: number of first receiver position (natural numbering)

ir_step: step in numbering of receiver points (e.g., 2 if every second receiver point has been occupied)
If ir_start and ir_step = 0, receiver station numbers remain unchanged

dt: trigger time correction [s]. If trigger is too late, i.e., the signal at 0m offset arrives before zero time, give a positive number, if signal arrives too late, give a negative number. If no geophone is placed at zero offset, you may use the air wave to check the trigger offset, perhaps after having applied a high-pass filter.

interp factor with which to increase the number of samples per trace (interpolation). I.e., if sampling interval is too large

receiver_corrections.dat

This file is used to correct geophone problems occurring in all recorded shots. You may only indicate geophone positions where there is really a problem, no need to have a line for each geophone position It has two columns:

igeo, fact

igeo: number of geophone position (natural numbering starting at one)

fact: factor with which to multiply the data. This factor may be
0 if the geophone recorded only noise and is eliminated from records
-1 if the polarity should be inverted.

If during interactive data treatment with program PyRefra.py amplitudes are changed or traces are muted, the program (when terminating normally) writes file "**receivers_modify.dat**". This file may be renamed and used directly as receiver_corrections.dat.

PyRefra.py checks in the beginning if one or both of these files exist and if needed applies corrections in the data structure: ipos, ir_start and ir_step are used to change shot-point and receiver-point numbers in the headers, fact is directly applied to the corresponding trace data; if dt is not zero, the following procedure is applied: if dt is negative, (-dt/sample-interval) samples are taken away at the beginning of the traces and the same number of samples is added at the end with zero values. If dt is positive, (dt/sample-interval) samples are added at the beginning with zero values and the same number of samples is taken away at the end of the traces; In this way, the traces maintain their initial length in number of samples.

Another optional file, **PyRefra.config**, may contain two text lines used for plot titles:

`title` (used as title for most windows)

`direction` contains the geographic direction of the beginning of the profile, may be one of [S, SSW, SW, WSW, W, WNW, NW, NNW, N, NNE, NE, ENE, E, ESE, SE or SSE].

The corresponding direction of the end of the profile is then determined automatically. So, if you enter e.g. SW for the beginning of the line, this means that the line runs in SW -> NE direction.

If this file does not exist at the beginning of a run, a dialogue box is opened asking for this information. The values are then stored in a newly created file PyRefra.config which will be used automatically the next time pyrefra is started.

Running pyrefra

To start pyrefra.py, open the Anaconda Powershell console, change folder to the one where your data are located (not absolutely necessary, but simplifies work), type

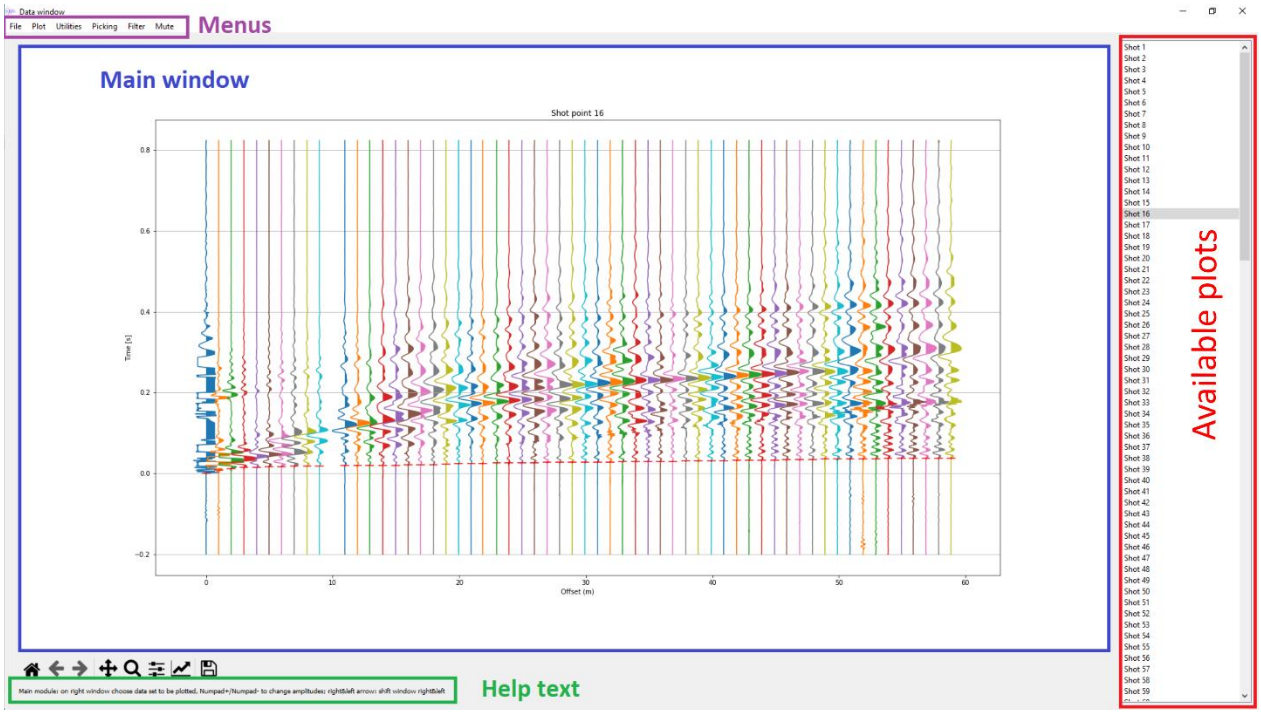
```
conda activate pg
```

and then simply type `pyrefra` followed by `ENTER`.

If file PyRefra.config does not exist in the data folder, the program will first ask for the title and direction (see above). If this file is found the information will be used. If the file content must be changed, do it in a text editor or erase the file and restart the program.

Then the program will ask for the files to be opened. It proposes initially files with ending “seg2”. If none is shown in the dialog box, probably you are dealing with Summit2 or SEGY files, so you may change the default ending to sg2, segy or sgy (or even show all files). Usually, you will want to open all available files. Click on one of them and chose them with CTRL-A. If not, you may choose them as usual in the explorer, using SHFT or CTRL keyboard keys. No problem, if also folders are chosen with CTRL-A, the program filters them out. If you know that there are bad files, you may copy them into another folder or change their name-endings to something else than “.seg2”, “.sg2”, “segy” or .”sgy” in which case the default option will not find these files and they are not opened. File numbering does not need to be contiguous. For information: if segy files are read, internally, the program copies the important segy header entries to seg2 headers. If the program has problems, it might be that not all necessary header entries exist or have been copied over.

When opened, the GUI shows in the main window the data of the shot gather corresponding to the first shot point. A smaller window to the right shows all available shot point numbers. Clicking on one of the names plots the seismograms of the corresponding shot point in the main window. Beneath the main window, a help text is shown. To be visible, it may be necessary that the Windows task bar be automatically hidden. Above the main window, a menu bar is placed.



Main module: on right window choose data set to be plotted. Numpad+/Numpad- to change amplitudes; right&left arrow: shift window right&left

The GUI is automatically sized to full screen size. The window size may be reduced, however, if the window is too small, certain graphs containing multiple partial windows (e.g., tomography results) will not show up correctly, even in the stored png files (overlapping axis texts of different sub-windows).

ATTENTION: in the following, if shortcut letters are given as capital letters, this is only for reading purposes. Usually, you will use small letters. Do not use SHFT for single letters, since the combination with SHFT has usually another meaning. However, if the caps lock has been pressed, this is ok.

Actions without Menu:

Numpad or standard + or – (on AZERTY keyboards without numpad, don't forget to press SHFT with the "+", which is not the case for other shortcuts)

Increase or decrease amplitudes of traces

Keyboard arrows right or left

If you have made a zoom in X (showing only part of the traces), the arrow right will shift the view to the traces further to the right (larger X coordinates), the arrow to the left will show traces with smaller X coordinates. The size of the window in terms of X-coordinates will always be maintained. E.g., your zoom shows traces 1 to 50 out of 120 traces. The Right-Key will show traces 51 to 100, a further click will show traces 71 to 120 (still 50 traces are shown, but since the last trace is n° 120, the window starts at 71). A Left-Key click will now show traces 21 to 70 and a further Left-Key click will show again traces 1-50.

The Menus

File Menu

Choose Data File (keyboard shortcut: CTRL-O)

Usually, you will not need this menu point. It allows you to choose new data sets.

Save SEGY (keyboard shortcut: Y)

Save data in SEGY format (e.g., to continue with Seismic Unix)

A dialog box opens where you may choose a series of options:

Data to be saved:

all data, including possible data recorded before the trigger

all data recorded after the trigger

only data visible on the screen (if zoomed)

data recorded after trigger, but first arrival signals are muted. This option makes only sense if picks have been measured. If this option is chosen, a second dialog window opens, asking for the approximate source signal length. The program searches for the minimum amplitude after the pick_time+signal_length and before pick_time+2*signal_length. Then it applies a slope of length signal_length/4 to the data before this minimum, bringing them gradually to zero. All data before this slope are muted. If no pick exists, it is supposed that the corresponding trace has strong noise and it is entirely muted. As a consequence, if the first arrival cannot be measured, but the later signal seems to be ok, a dummy pick should be placed (save perhaps first the file picks.dat with the useful picks to another name to be able to do tomography). The data of traces without pick are saved in the file without any mute and the header value duse is set to 0 (normal value is 1). In this way, those data may be excluded from treatment in SU with the following command:

```
suwind <in_file.su key=duse min=1 >out_file.su
```

Data of all shots or only those of the active shot

Save all data in one single file (this file will be called prefix00000.sgy) or each shot in another file

Multiplicator for distance: distances are stored in SEGY as integers. If you need the cm, set multiplicator to 100 (default value).

Store multiple receivers: What to do if the same combination shot-point receiver-point has been recorded. By default, only the first occurrence is store, following traces are ignored.

If a frequency filter had been applied before, you may apply this filter to all traces.

Save SU (keyboard shortcut: U)

Save data in SU format (e.g., to continue with Seismic Unix). The same dialog box as for SEGY storing appears. It seems that in SU format, several header words are not correctly filled in by obspy. Therefore, in my experience, it is better to save the data in SEGY format and use the following command in Seismic Unix (Linux) in order to erase the file header and pass to SU format:

```
segyread tape=file_name verbose=1 | segyclean > data.su
```

Save SEG2 (keyboard shortcut: 2)

Save data in SEG2 format (e.g., if you have done quite a few corrections via file_corrections.dat or receiver_corrections.dat and/or a number of files were eliminated, and you want a new list with continuous file numbering). You may save just the gather on the screen or all available gathers (into a single file or every gather into its own file). You may write file-gathers or shot-gathers. If only the gather on the screen is to be saved, only the corresponding type of gather may be written, however, receiver gathers and distance gathers cannot be stored, and a warning message appears. File gathers are written into the interactively chosen folder as “file_nnnnn.seg2” and shot-gathers as “shot_nnnnn.seg2”, nnnnn being the renumbered file number (starting at 1) or the shot number.

Save FWI (no keyboard shortcut)

Save only data in binary format without any header (4 bytes)

Save ASCII (no keyboard shortcut)

Save data in ASCII format

Line 1: t0, dt, ntrace, nsamp, maximum value, factor, x_source [m], y_source [m], z_source [m]

t0: time of first sample [s]; dt: sampling interval [s]; ntrace: number of traces; nsamp: number of samples per trace

Line 2: ntrace columns containing the value of each trace corresponding to “maximum value” of line 1

The amplitudes are calculated as stored_value/(maximum_value*factor)*corresponding_value “stored_value”: Integer values stored in the rest of the file; “corresponding_value” value in this Line 2 for each trace

Lines 3 to 5: X_receivers, Y_receivers, Z_receivers (each line ntrace values)

Following nsamp lines: Normalized values of all traces (one line per time interval, one column per trace)

Save headers (keyboard shortcut: H)

Writes headers of all traces to ASCII files. Every data file has its corresponding header file "header_nnnnn.dat". These header files are written to folder Headers. If this folder does not exist, it is created.

Save Plot (keyboard shortcut: P)

Save actual screen into png file. File names depend on what is actually shown

Quit (keyboard shortcut: CTRL-Q)

Finish program with cleaning up. A dialog window opens to confirm termination.

Display Menu

Original data this screen (keyboard shortcut: O)

Plot the original, not treated data shown on the actual screen (e.g., to undo filtering). All other data are not reset. A potential zoom is maintained.

Original data all (keyboard shortcut: SHFT-O)

Plot the original, not treated data of all files (e.g., to undo airwave muting or velocity filter applied to all files). A potential zoom is maintained.

Shot gather (keyboard shortcut: S)

Plot shot gathers (in contrast to file gather), i.e., all traces from all acquired files that have been recorded from the same shot point position are plotted into one record section. If the same combination receiver point – shot point has been recorded several times, the traces are plotted overlapping. The list of plots on the “Available plots” window is updated and the gather corresponding to the first shot is shown in the main window. This is the default plot when starting the program.

Plot data of one file (keyboard shortcut: F)

Plot file gathers (in contrast to shot gather), i.e., the traces from the acquired files (one file = one shot) are plotted into one record section. The list of plots on the "Available plots" window is updated and the gather corresponding to the first file is shown in the main window.

Receiver gather (keyboard shortcut: R)

Plot receiver gathers, i.e., all traces from all acquired files that have been recorded at the same receiver position are plotted into one record section. If the same combination receiver point – shot point has been recorded several times, the traces are plotted overlapping. The list of plots on the "Available plots" window is updated and the gather corresponding to the first receiver is shown in the main window.

Distance gather (keyboard shortcut: D)

Plot distance gathers, i.e., all traces from all acquired files that have a common absolute offset are plotted into one record section. If the same offsets have been recorded several times, the traces are plotted with a slight lateral shift. The traces are plotted at the mid-point position between shot and receiver. The list of plots on the "Available plots" window is updated and the gather corresponding to the smallest offset is shown in the main window. This window may serve for quality control of the picks. Especially, in the case when receiver and shot point are switched both traces are plotted at the same position and should have the same arrival time.

Choose component (keyboard shortcut: G (for "geophone type"))

If several components have been recorded, choose the one to be plotted or "all" for plotting all components

Phase_angles (no keyboard shortcut)

If several components have been recorded, plot the angles (arctan) between full horizontal component and vertical component (inclination) and between the two horizontal components (if they exist) (declination). The first trace at one position is inclination, the second declination.

TODO: For the moment, when clicking this option, only the hook is set, but the plot is not changed immediately. The next time you choose a file/shot/receiver to be plotted, the angles will be plotted.

Zoom options:

Most zoom options are maintained from one plotting window to the others

Zoom (keyboard shortcut: Z)

Trace a rectangle with left mouse key maintained pressed to select data to be shown (as well in time as in distance). When the mouse button is released, the zoom is accepted. The time limits of the zoom will be maintained from one plot to the next, however not the x limits. When calling a new plot, again all traces are plotted.

Zoom out (keyboard shortcut: CTRL-Z)

If several zoom levels have been defined using "Zoom", return to the next smaller zoom.

Zoom in (keyboard shortcut: SHIFT-Z)

If several zoom levels have been defined using "Zoom" and you made a Zoom-out (or Zoom-initial), return to the next tighter zoom.

Zoom initial (keyboard shortcut: ALT-Z)

Go back to the original plotting limits, i.e., show all data.

Gain options:

Gain options are maintained from one plotting window to the others.

Trace normalize (no keyboard shortcut)

Plot the traces normalized to the maximum value of each trace (default setting).

Time gain (no keyboard shortcut)

Apply a time gain (traces are in addition trace normalized). A dialog window opens, asking for the exponent of the time gain, i.e., amplitudes are multiplied by (t**time_gain).

Distance gain (no keyboard shortcut)

Apply a distance gain (traces are **not** trace normalized). A dialog window opens, asking for the exponent of the distance gain, i.e., amplitudes are multiplied by (x**distance_gain).

AGC (no keyboard shortcut)

Apply an Automatic Gain Control (traces are in addition trace normalized). A dialog window opens, asking for the AGC window length.

Utilities Menu

P_Model (keyboard shortcut: ALT-P)

Allows to construct a 1D velocity model by tracing a series of straight lines across the record section. To trace a line, press left mouse button and pull the line. Releasing the mouse button is accepting the line.
ATTENTION: The first mouse click determines on which side of a shot point you want to draw the line. It is not allowed to cross the zero-offset. This is even valid if the shot point is on one end of the line. In this case, do not make the first click outside of the seismogram section (negative side if the shot point is at the left, positive side if it is at the right.
The first line on each side of the shot point is considered to be the direct wave and is forced to pass through the origin. For all further lines, the position where the mouse is clicked is one point of the line, the position where the mouse is released is the second point.

When releasing the mouse, a dialog window opens which allows you to accept the line and trace the next one, to accept it and finish the model or to redo the line. In addition, when clicking on "Show details", the velocity, and intercept times are given as well as the depth to the layer limit (refractor) and the thickness of the layer above the refractor.

The lines plotted will disappear when a new plot is called (right window or different gather). A new call to P_Model creates a new model, starting with the direct wave.

When the model is finished, it is written to a file in the data folder with the following name structure: prefix_number_date_time.1D**P** where prefix may be "shot_point", "receiver_point" or "file" depending on the type of gather used (it is supposed that the routine is not used when a distance gather is shown).

The resulting models are not further used in the inversion process. They are mainly meant for teaching purpose to allow students to get a rapid idea of the velocity structure and its lateral variability below a profile.

S_Model (keyboard shortcut: ALT-S)

Same as P_Model only that the model is considered to be an S-wave velocity model. The results are stored in a file named prefix_number_data_time.1D**S**. See P_Model for more explanations

Tomography (keyboard shortcut: T)

This tool is only activated if measured travel times exist. It uses Pygimli to invert travel times into a tomography model. If Pygimli has not been installed, the option will never be activated.

On call, a dialog box is opened asking for:

Maximum depth of the model [m]: The default value is calculated as 1/3 of the maximum available offset.

Initial smoothing parameter: The bigger the value, the smoother will be the model. This will be a parameter to play with, doing different inversions with different smoothing. If the smoothing parameter is negative, the absolute value will be used for initial smoothing and the reduction of this parameter during the iterations will be automatically optimized and the following parameter (Smoothing reduction) will be ignored (not clear though, how PyGimli handles this…).

Smoothing reduction per iteration. If not zero, the smoothing parameter will be multiplied by this factor after each iteration.

Smoothing in z direction: This value may be between 0 and 1. When 0, the different layers of the tomography will be independent from each other, if 1, the same smoothing factor will be used in vertical and horizontal direction.

Maximum iterations: The number of iterations may be limited with this value, e.g., for testing purposes. In general, the iterations are stopped if the data fit does not become better anymore, i.e., if the chi^2 value does not decrease by more than 1% of its actual value or if chi^2 becomes smaller than 1.

Initial and final velocities: Ray tracing needs an initial model with a vertical gradient. This initial model is defined by these two values. **TODO**: allow for a refined starting model.

Minimum and maximum allowed velocities: Pygimli allows to limit search space of model velocities. Give here the desired extreme velocities. **TODO**: Understand working of Pygimli: Even if very large upper velocity limit is given, pygimli limits the velocity model to a range of values much narrower than given. Only if the parameter “limits=[vmin,vmax]” is not at all used, velocities seem to be completely free. Therefore, if both given velocities are zero, the program makes a call to mgr.invert without using key word “limits”.

Velocity color scale min and max: If you do inversions for different profiles, it may be interesting to have the same color scale for all. In this case, you may give the values of the minimum and maximum velocities appearing in the color bar.

If min or max are =0, the corresponding limits of the color scale are calculated automatically using the 1% and 99% percentiles of the obtained velocities.

Type of color scale: The user may choose between different color scales. A special color scale has been created for P-wave models which plots the 1500 m/s velocity by cyan color, in order to highlight an approximate upper limit of the fully saturated zone. For S-wave models, a similar color scale highlights the 500 m/s velocity by cyan.

Plot title: The default title is the one given in file PyRefra.config (see "Data preparation"). You may change it to any text. Appears only in main title of the inversion results plot.

At the end of the inversion, the resulting model is shown in the upper 60% of the screen. Below, smaller sub-windows show the initial model, the rays of the final model with the "coverage" (i.e., a measure of resolution), the measured travel times as function of shot point position and receiver point position, the misfits in a similar plot calculated as "calculated times minus measured times" and finally, a plot showing average misfits for the different shot points and the different receiver points. This last plot allows you to verify if there are problems with certain shots or receivers.

This plot is stored in a png file, together with other inversion results such as velocity model, misfits, rays of final model and chi²-evolution in a folder created automatically by Pygimli named ./date-time (where the "." is the data folder; date and time of folder creation).

If you are not happy with the colour scale or you want to change the maximum plotted depth, you may press "c" or "C" (but not "SHFT+c") to call a new dialogue window asking for this information. See also below "Change colors tomo". Before doing this, it may be advantageous to increase the window size to full screen (using the standard icons) in order to avoid overlapping axis labels between the subplots.

The program offers writing output files for use in IFOS2D (https://git.scc.kit.edu/GPIAG-Software/SOFI2D) or ShaVi (https://github.com/vatsvikki/ShaVi-1.0), both software for FWI. For IFOS, model files for P-waves, S-waves and densities (see function prepareSOFI for more information), receivers and shots in SOFI format as well as Jason-format control file are stored. For ShaVi, data are stored in ASCII format, they may be reduced in terms of sampling rate, recording length and number of shot points. Also shot and receiver positions are stored, a file with Ricker source signal is created as well as a control file. These files are stored in a folder "ShaVi" which will be created if it doesn't exist.

To leave this plot, close the window or simply go back to the main screen. To show the plot again, press "C" again.

Checkerboard test (keyboard shortcut: ALT-T)

In order to do the checkerboard test, picks must have been stored as "picks.sgt" file. The information in this file will be used for the measurement configuration (positions of shots and receivers, which picks exist). This file may be created by calling "Tomography" or through option "Picking -> Store Gimli format".

First, a checkerboard model is created. The dialog box asks for the following information:

Maximum depth of model (for the moment, only horizontal surface at z=0 is allowed)

Horizontal size of blocks [m] (by default 1/5 of horizontal line length is proposed)

Vertical size of blocks [m] (by default 1/15 of horizontal line length is proposed)

Starting position X: The first block may start at the beginning of the line, but it may also start e.g. with a half-block. The value given is in % of horizontal block size. Example: Block size is defined as 10m. If Starting position X is set to 20, the first block has a size of 2m, followed by 10m blocks and the last block goes until the end of the line.

Starting position Z: Similar to Starting position X, only for vertical positioning of blocks

Initial velocity at surface, initial velocity at bottom, both in [m/s]: These values define the background model. Different velocities allow defining a vertical velocity gradient.

Velocity difference: The checkerboard velocities are defined by local differences with respect to the background model. These differences may be defined relative or absolute. If the given value (let's call it diff) is smaller than 1, relative differences are supposed, i.e. v1 = v_background*(1-diff) and v2 = v_background*(1+diff). If diff is larger than 1, it is supposed to indicate an absolute difference, i.e. v1 = v_background-diff and v2 = v_background+diff

Noise level: Noise level to be added to the calculated times in [s].

The program then calculates the synthetic model and the travel times for the combinations shot/receiver found in picks.sgt.
Then a second dialog box appears where certain inversion parameters may be set as for the tomography option (see there for explanations.
Finally, a window opens showing at the left top the synthetic model, at the left bottom the result of the inversion and at the right synthetic travel times as crosses and travel times resulting from the inverted model as continuous lines.

Envelopes (keyboard shortcut: E)

Tool calculates and shows Envelope of all traces. For the moment, only the plot is shown. Option exists only for testing purposes, therefore this option is inactive, but may be activated in one of two ways: by editing the file refraWindow.ui, e.g., using QtDesigner, or, by uncommenting the line "self.window.Envelopes.setEnabled(True)" near the end of the __init__ of class Main (file PyRefra.py). The results for all traces of the treated shot point are stored in file envNNNNN.asc (NNNNN is the shot point number) the contents of which should be self-explaining.

Tau-P (keyboard shortcut: CTRL-T)

Tool calculates and shows Tau-P analysis. For the moment, only the plot is shown. If you want to save it, use File -> Save plot. **TODO** use obspy function for this.

False colour (keyboard shortcut: SHFT-F)

ATTENTION: This option is purely experimental and is not meant for use in data treatment! The idea behind is to find out whether certain combinations of indicators would allow a better automatic picking. If you are interested to pursue this possibility, you are invited to add further options in function "falseColour" located in file refraData.py

Possibility to plot 1 to 3 different indicators that may help for picking. This tool is for the moment only meant for display, to analyse data visually. Possible indicators are: Instant frequency, envelope, 2nd derivative of envelope, 2nd derivative of data, Sta-Lta transform, Akaike Information Criterium (AIC), max-min relation and autocorrelation. Max-min relation plots the relative amplitude difference between a maximum and its following minimum on the one hand and the same maximum and its preceding minimum on the other hand. Autocorrelation is usually plotted on its own since the time scale has no relation with the other options. On top of the autocorrelation, two traces are plotted: white is the average autocorrelation, yellow the average trace obtained from averaging the frequency spectra. A dialog box will open where you may check the desired indicators (1, 2 or 3 out of the 8 possibilities). The first indicator you click on will determine the red channel, the second one the green channel and the third one the blue channel. This implies also that if only one indicator is chosen, the plot will be between black and red. If 2 are chosen, colour may change between black, red, green, and yellow (no blue). If picks have been measured for the analysed gather, they are plotted on top of the false colour plot. In addition, a magenta line is plotted indicating evolution of maxima of the combined indicators (maximum brightness, Pythagoras of color values), as well as a white line which represents an optimum fit of the positions of the maxima via two straight lines. The maxima are possible picks, the straight lines indicate a possible two-layer velocity model. These lines are though not always meaningful.

Change trace sign (keyboard shortcut: I)

You may choose one or more traces which have wrong polarity (geophone or receiver electronics may have been reverse-cabled). Left click on one trace after the other. Each chosen trace is marked by a small red star plotted near the bottom of the trace plot. When all traces have been marked, press right mouse key.
When the program is finished using “File -> Quit”, a file “receivers modify.dat” is written indicating the muted traces, which may eventually be renamed to “receiver_corrections.dat” to make these changes durable.

Change sign (keyboard shortcut: CTRL-I)

Depending on the polarity of the acquisition system, the general polarity may be reversed. In order to show first arrivals in positive direction (filled black), you may use this tool that changes the sign of all recorded traces, not only those visible in the actual window.

Change colors tomo (keyboard shortcut: C)

This option is only activated when a tomography model has been calculated. It allows plotting the results again and, if wanted, changing colour scale of model and depth extent shown.

Wave animation (keyboard shortcut: W)

Plots animation of wave evolution in time along the geophone line actually on the screen. Animation goes from first to last sample of actual time zoom. Data are smoothed with 3-point running average and amplitudes are cut at 99% quantile.

Attenuation (keyboard shortcut: Q)

Function searches for each trace the maximum of the envelopes of data plotted on the screen (you may use muting functions to focus surgically on certain phases). Amplitudes are multiplied by the absolute offset to counteract geometric spreading. Then, for each side of a shot point, an exponential function is fitted to the amplitude evolution, if at least 4 traces are available. If more than 6 traces exist on the corresponding side, two independent lines are fitted whose results may be interpreted as attenuation near the surface and deeper down. A plot is presented with the fitted logarithm of the amplitudes and the amplitude fit itself. Instead of slope, a Q value is indicated (-1/slope) as well as a r^2 value for the ensemble of the two lines.

All results, also from different shot points, are stored in file Q.dat

Pseudo_velocity (keyboard shortcut: V)

Function produces two figures on one screen: First the average velocity (pseudo velocity) for every pick as function of offset (y-axis) and midpoint position between shot and receiver (x-axis). In the second plot, local slownesses are plotted, as well as function of offset and midpoint. For this calculation, for every shot point, at receiver i, the slowness is calculated as :
(t[i+1]-t[i-1])/(offset[i+1]-offset[i-1]). In this way some smoothing is done. If one of the times does not

exist (trace i-1 or i+1 has not been measured), the slowness value is stored as nan. The Y axis is scaled with offset/3, which gives a very (very) rough estimate of depth. For slownesses, a logarithmic colour scale is used. The plot is stored in file “`pseudo_section_slowness.png`”. This kind of plot may serve to detect problems with single measurements or shots/receivers.

Picking Menu

Introductory remarks: try to pick as many of your traces as possible before applying any filter or other data treatment. This is especially true for traces near the shot point, which have in general still high frequencies such that their waveform is often strongly affected by frequency or velocity filters.

From my experience, the best automatic picking results are mostly achieved using correlation picking, using it first on a good shot or file gather or doing first manual picking on one gather.

Manual picks (keyboard shortcut: M)

Manual picking. For all picks, two clicks with the **left mouse button** are needed: first you mark the position of the pick, then the uncertainty, which is considered to be symmetric around the pick position, so you may click above or below the pick position. A **double click** defines as default uncertainty 2 time-samples if no filter was applied to the data and 4 samples if a filter was applied.
To finish, click right mouse button
You may **erase picks** by clicking the **central mouse button (or the mouse wheel)**. The program searches first the trace nearest to the mouse click and erases then the pick nearest to the click. If no pick is available for the corresponding trace, nothing will happen.
Right mouse button finishes picking.
When calling the tool, a stippled line is plotted showing the theoretical air-wave arrivals, which allows avoiding picking this wave as apparent direct wave, when working with near-surface data. If picking has been done on other shots, also a green stippled line is plotted, indicating picks done on other nearby traces at the same offset (maximum distance of nearby trace: 10 traces). Clicking **SHIFT+left mouse button** uses these picking times as picks for the actual gather and finishes picking. Picks may then be replaced using Move Picks (CTRL+M).

Amplitude picking (keyboard shortcut: CTRL-A)

Automatic picking tool that locates first arrivals by analysing full amplitudes. First all maxima and minima are determined. For each maximum (minimum), the function calculates first an amplitude parameter corresponding to amp_pos = max-(min1+min2) for maxima and amp_neg = max1+max2-min for minima, where 1 means the next extreme value before the actual one and 2 the next extreme value after the actual one.

Then a vector is calculated with the relative amplitudes, i.e. dmax[i] = amp_pos[i]/amp[i-1] and dmin = amp_neg[i]/amp_neg[i-1]. The pick is located in a first try near the highest value of dmax, located later than a time “tstart”, precisely at the position of this maximum minus half the distance between the corrresponding maximum position and the following minimum position. “tstart” corresponds to the virtual arrival time of “maximum velocity” asked for in the dialog box that opens on call. If the maximum value of dmin is earlier than that of dmax and its value is larger than the one of dmax, then the pick is switched to the position of maximum dmin, shifted again by half the distance between this minimum and the following maximum.

Then, as a last test, the function searches for a (near) zero-crossing between the chosen pick position and the corresponding maximum (minimum) of dmax (dmin). If there is one, it will be the final pick, if not the chosen pick is maintained.

Picking STA-LTA (keyboard shortcut: SHFT-P)

Standard STA-LTA picking. You must give window lengths for STA (short) and LTA (long). You have the possibility to do picking for all traces of the record section or first do a zoom and calculate picks only for the visible traces.

Correlation picking (keyboard shortcut: SHFT-C)

Do first one manual pick on a good trace which will be used as reference trace. From this pick on, further picks are searched by cross correlation. If picks from other shots exist, they are used by the algorithm as guide (maximum correlation near to other picks made at the same offset). The picks are only calculated for the traces visible on the screen. So, if part of your data is noisy, you may first make a zoom on the good traces, do the correlation picking, then filter the data and do another zoom on the traces not yet picked. Uncertainties are set to 2 samples if no filter was applied to the data, else to 4 samples.

Move picks (keyboard shortcut: CTRL-M)

Move picks to other position. Especially useful to adjust automatically done bad picks.
Click left near the pick to be moved. The vertical arrow keyboard keys shift the pick by one sample upward or downward. Arrow key pressed with SHIFT displaces picks by 10 samples, with CTRL, shift picks by 1ms.
When pick is at the correct position, click left on the next pick or use horizontal keyboard keys to go to next trace to the left or to the right.
To finish, click right mouse button

Uncertainty change (keyboard shortcut: SHFT-U)

Change uncertainties of picks. Use is similar as for moving picks: Vertical keyboard arrows increase or decrease uncertainties. Minimum uncertainty is set to 2 time-samples. Finish with right click on mouse

Erase all picks (keyboard shortcut: CTRL-E)

Erase all picks of the actually shown seismogram gather (other picks are maintained). Usually used if automatic picking gave too bad results.

Plot all picks (keyboard shortcut: CTRL-P)

Plots all picks at the coordinate of the corresponding trace. Picks are colored by shot point number.

Plot calculated times (no keyboard shortcut)

Activated after using Tomography or if the program at start finds a file called “calc_picks.dat”. The calculated picks are plotted as a connected line. As long as this tool is not pressed again, calculated picks are plotted for all gathers. Deactivation of the tool takes only effect at the next screen refreshment (next gather, next zoom etc.)

Store picks (keyboard shortcut: CTRL-S)

Store all available picks (not only those of the actual seismogram gather) into file “**picks.dat**”. It has 5 columns:

nr shot point, nr receiver pt, picked time, picked time minus uncertainty, picked time plus uncertainty nr shot point as in file shots.geo, nr receiver point as in file receivers.geo. The uncertainties are normally considered to be symmetric, but certain automatic picking routines may give different uncertainties before and after the pick.

PyRefra stores automatically all picks each time picking or modifications of picks are finished with right mouse key.

Store Gimli format (no keyboard shortcut)

Stores calculated picks in Gimli format (file **picks.sgt**), used by the Tomography tool. Usually, you do not need this tool, since when calling Tomography, the measured picks are automatically stored in this format before starting inversion.

ATTENTION: If your working directory is on an external disk, it seems that losing the contact once is losing it until you restart the program: Therefore, if “Store Picks” or “Store GIMLI” detect an error the program is stopped, and you lose only the data not yet stored since the last CTRL-S or manual picking.

Filter Menu

Frequency (keyboard shortcut: CTRL-F)

Frequency filter. Program calculates average FFT of all shown traces and allows user to define corner frequencies. Click left mouse at position of low-cut frequency, pull line and release at high-cut frequency. If one of the two frequencies is clicked outside the frequency axes (negative frequency or more than maximum frequency), the corresponding fllter is not applied.
After filter frequencies have been defined graphically, a dialog box allows modifying them (e.g., rounding).
If a frequency filter had already been defined before, the dialog box opens immediately, proposing the frequencies of the last applied filter. You may accept, change them manually or ask for graphical redefinition.

Frequency filter for single trace (no keyboard shortcut)

Same as standard frequency filter, but you must choose one single trace to filter, by left clicking onto it.

Air wave fk (keyboard shortcut: SHFT-A)

F-K filtering of all velocities less than 400 m/s.

Velocity filter (keyboard shortcut: CTRL-V)

F-K filtering choosing interactively the cutting velocity (all smaller velocities will be attenuated).
The cutting velocity is defined interactively by pulling a slider to the desired position. To accept the chosen velocity, release the mouse. The program then asks you whether you want to apply the same filter to all gathers or only to the visible one. The chosen velocity is saved as default velocity for further velocity filtering.

Mute Menu

Mute/recover trace (keyboard shortcut: CTRL-T)

Click left on all traces you want to mute (small red star is plotted near the lower end of the trace) or which you want to plot again if it was muted (a small green star is plotted at that place). Finish with right click.

The muting will be applied to all traces having been recorded at the same receiver position.

When the program is finished using “File -> Quit”, a file “receivers modify.dat” is written indicating the muted traces, which may eventually be renamed to “receiver_corrections.dat” to make these changes durable.

Mute air wave (keyboard shortcut: A)

Mute a stripe around the air wave (mainly thought as preparation before treating data as reflection seismics). In the dialog box give estimated air-wave velocity, the total with of the stripe and the width of the taper to both sides. If you are not happy with the result, go back to original data (SHFT-O) and try again.

Mute before line (keyboard shortcut: L)

Click left and draw multiple line segments. The last segment is finished with right click (i.e. the right click defines one more point). Data before the traced line are eliminated for each trace before the nearest zero-crossing. Traces that are not crossed by the line (at the right or left side of the screen will not be muted!

Mute after line (keyboard shortcut: SHFT-L)

Click left and draw multiple line segments. The last segment is finished with right click (i.e. the right click defines one more point). Data after the traced line are eliminated for each trace from the nearest zero-crossing on. Traces that are not crossed by the line (at the right or left side of the screen will not be muted!

Example data:

For the data in the given example, 60 geophones were placed at 1m distance from 0 to 59m. The GPS coordinates rotated and shifted to a local coordinate system starting at 0m are found in file “receivers.geo”. 31 Shots were fired every 2 m from 0 to 60m. Their local coordinates are found in file “shots.geo”.

The numbering of the files is not contiguous, because several shot were repeated and the bad files were deleted. So, although the file numbers go from 1 to 34, only 31 files are stored, one per shot point.

Several shots had bad trigger timing, triggering too early. In file “file_corrections.dat”, the time corrections are given for recorded files (not shot point numbers!) 6, 7, 8, 9, 11, 14, 18, 21, 25 and 26. In recorded file 23, a wrong shot point number was stored. This is also corrected by an entry in file “file_corrections.dat”. Not corrections for specific receivers had to be applied, therefore, no file “receiver_corrections.dat” is present.

Finally, there is a file “picks.dat” containing picks already done by myself and a file “calc_picks.dat” containing calculated picks from a tomography model.

To test the program, you should copy all files into a folder which may be different from the program folder.

Now, you may first store files file_corrections.dat, picks.dat and calc_picks.dat somewhere else (or change their names), just to see the original data with the wrong arrival times for the above-mentioned files.

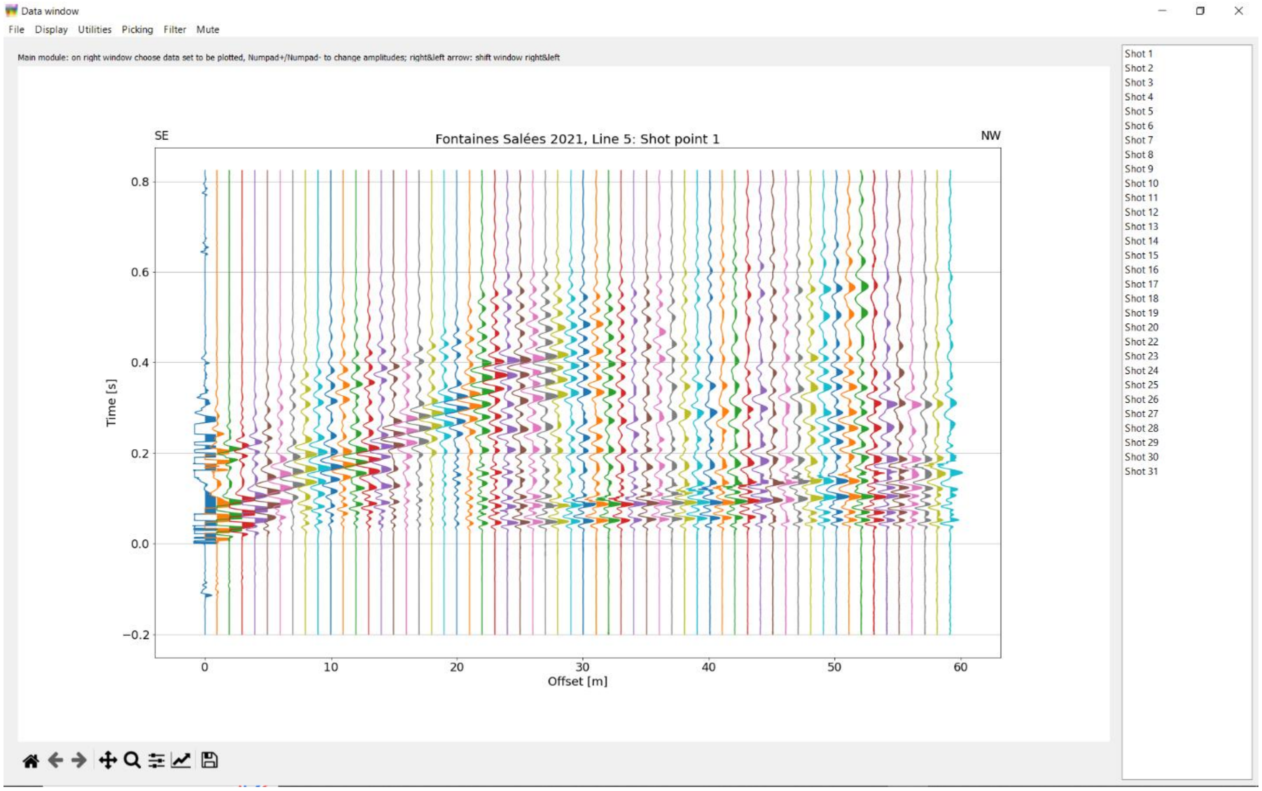
Edit PyRefra.py: Find line starting with `sys_path =` (approx. line 97) and replace the existing path with the path to the folder where the program files are stored. On the following line starting with `dir0 =`, you may replace the existing path with the one where the data files are stored.

Open spyder and within spyder PyRefra.py and run the program

Or open a command window (under Windows: Anaconda Prompt) and execute the following commands:

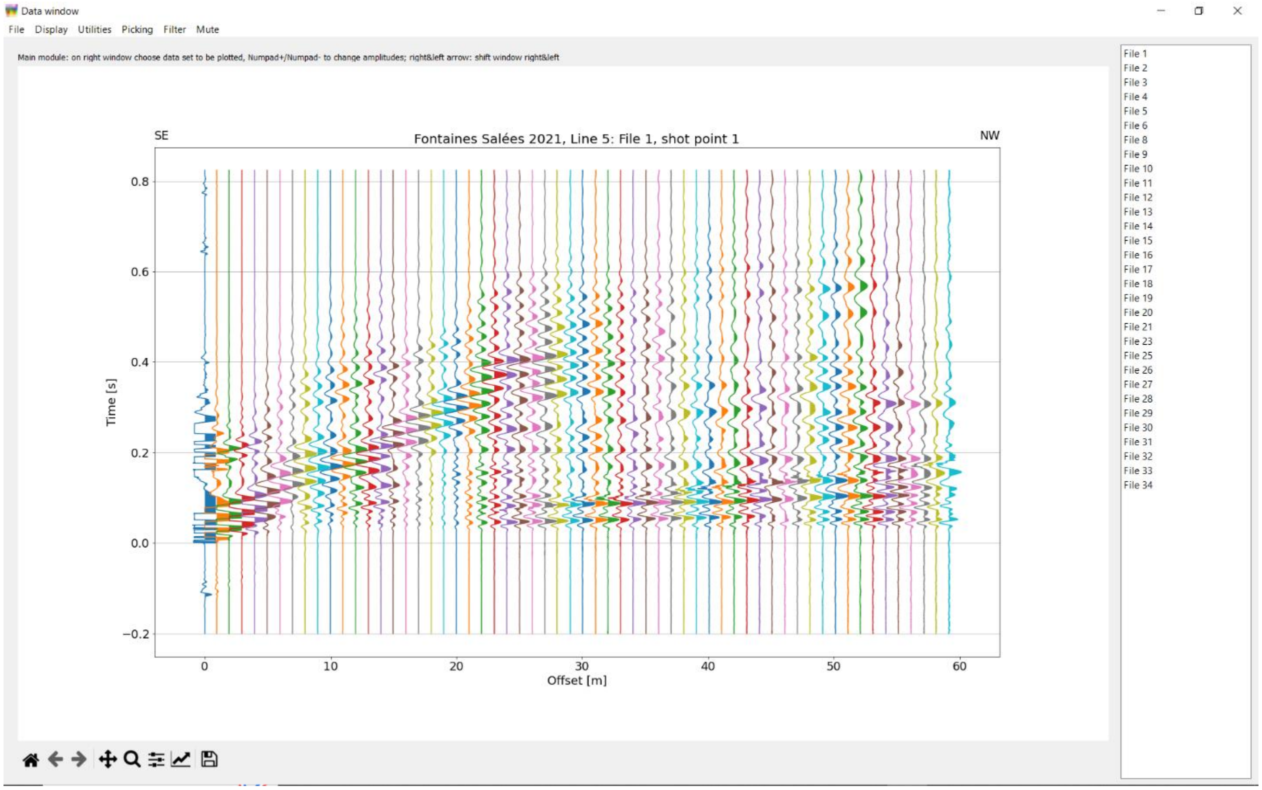
```
cd to the folder containing the program files
conda activate pg
python PyRefra.py
```

The program starts asking for the data files you want to treat. Chose all available ones. Then the data from the first shot point show up, which correspond to those of file 1. Shot points 1 to 31 are available in the right window.



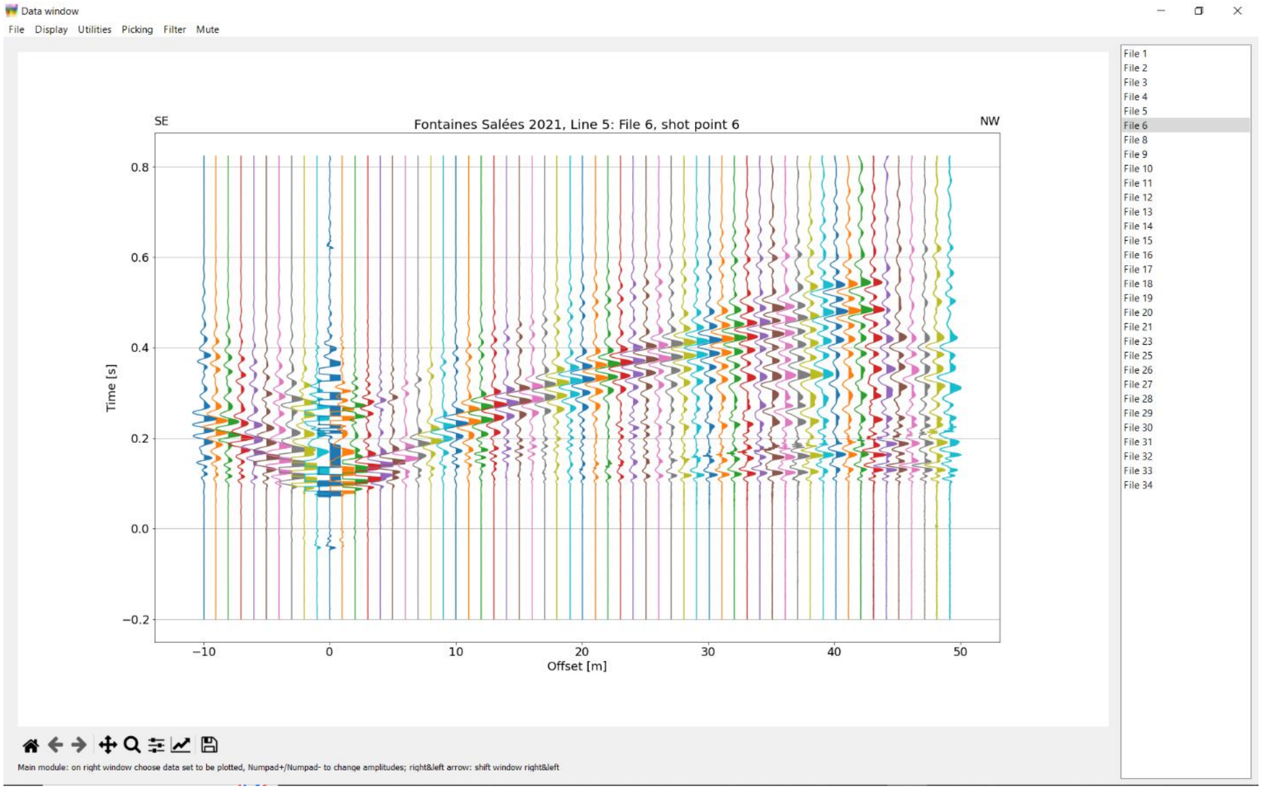
Main module: on right window choose data set to be plotted, Numpad+/Numpad- to change amplitudes; right&left arrow: shift window right&left

If you switch to file gather instead of shot gather, you will see that the available numbers change from 1 to 34, with a few missing numbers (missing files).



Main module: on right window choose data set to be plotted, Numpad+/Numpad- to change amplitudes; right&left arrow: shift window right&left

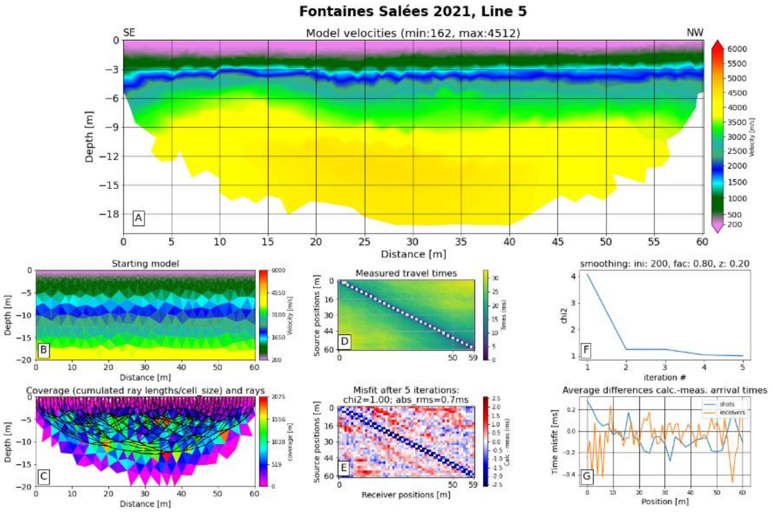
Click on File 6 in the right window and see the wrong trigger time:



Main module: on right window choose data set to be plotted, Numpad+/Numpad- to change amplitudes; right&left arrow: shift window right&left

Stop the program, copy file “file_corrections.dat” and “calc_picks.dat” (perhaps also “picks.dat”) back to the data folder and run again. The trigger time corrections are done, and you may explore the program and do some picking, creating a new file “picks.dat”. Try also plotting the calculated arrival times.

A tomography using the existing file “picks.dat” and the default options should give the following image except for the title (path to your data folder should appear):



Known bugs:

- Sometimes, the progress bar gets stuck, although the program continues working (this is always the case if you go to another window while the progress bar is working). Look for messages in the command window or in Spyder to see the real advancement. Anyhow, if possible (two screens) leave the command window visible to see if some error has occurred if program seems to be stuck.
- Sometimes, when doing manual picking, existing nearby picks are not shown.
- Sometimes the help line is written on top of the main window instead of at the bottom.
- Sometimes, the zoom is not behaving correctly. Go back to earlier zoom (CTRL+Z) and try again, it will work.